

1 1. In a networked environment, wherein one or more client computer systems
2 make requests for information from a server computer system, the server computer system
3 providing information in response to the requests from the one or more client computer
4 systems, the server computer system having one or more listen sockets and having a
5 backlog queue for queuing connection requests that the server computer system cannot
6 currently handle, a method of reducing denials of service even though the server computer
7 system is experiencing a denial of service attack, the method comprising:

8 attempting a connection for each connection request received by the server
9 computer system from said one or more client computer systems;
10 for each connection request that the server computer system cannot
11 currently handle, placing the connection request in a backlog queue;
12 monitoring the backlog queue;
13 determining that the backlog queue is being used;
14 resetting one or more connection sockets upon notification that the backlog
15 queue is being used.

16
17 2. The method in accordance with Claim 1, further comprising mapping each
18 connection request to a corresponding listen socket.

19
20 3. The method in accordance with Claim 2, wherein each listen socket has a
21 corresponding backlog queue.

1 4. The method in accordance with Claim 3, wherein placing the connection
2 request in a backlog queue comprises placing the request in the backlog queue
3 corresponding to the listen socket that the connection request mapped to.

4

5 5. The method in accordance with Claim 1, wherein attempting a connection
6 for each connection request received by the server computer system from said one or more
7 client computer systems is performed using a Winsock module.

8

9 6. The method in accordance with Claim 1, wherein attempting a connection
10 comprises calling a module that accepts connections and waits for request data before
11 completing.

12

13 7. The method in accordance with Claim 6, wherein the module that accepts
14 connects and waits for request data before completing comprises a Winsock()AcceptEx()
15 module.

16

17 8. The method in accordance with Claim 1, wherein monitoring the backlog
18 queue comprises calling a module that scans at least the backlog queue for activity.

19

20 9. The method in accordance with Claim 8, wherein determining that the
21 backlog queue is being used comprises detecting that the module that scans at least the
22 backlog queue has returned.

1 10. The method in accordance with Claim 8, wherein the module that scans at
2 least the backlog queue for activity comprises a Winsock()select() module.

3

4 11. The method in accordance with Claim 10, wherein determining that the
5 backlog queue is being used comprises detecting that the Winsock()select() module has
6 returned.

7

8 12. The method in accordance with Claim 1, wherein resetting one or more
9 connection sockets upon notification that the backlog queue is being used comprises the
10 following:

11 identifying any connection sockets that have connections but no received
12 request data; and
13 disconnecting the identified connection sockets.

14

15 13. The method in accordance with Claim 12, wherein identifying any
16 connection sockets that have connections but no received request data comprises the
17 following:

18 calling a module that identifies the state of the connection socket.

19

20

21 14. The method in accordance with Claim 13, wherein the module that
22 identifies the state of the connection socket comprises a Winsock()getsockopt() module.

23

24 15. The method in accordance with Claim 1, further comprising:

1 specifying a grace period between the time the backlog queue is determined
2 to be used and the time one or more connection sockets are reset to allow the server
3 computer system to empty the backlog queue, wherein the resetting of the one or
4 more connection sockets is performed only if the backlog queue still has entries
5 after the grace period.

6

7 16. The method in accordance with Claim 1, wherein attempting a connection
8 for each connection request received by the server computer system from said one or more
9 client computer systems comprises establishing a connection.

1 17. A computer program product for use in a networked environment, wherein
2 one or more client computer systems make requests for information from a server
3 computer system, the server computer system providing information in response to the
4 requests from the one or more client computer systems, the server computer system having
5 one or more listen sockets and having a backlog queue for queuing connection requests
6 that the server computer system cannot currently handle, a computer program product for
7 implementing a method of reducing denials of service even though the server computer
8 system is experiencing a denial of service attack, wherein the computer program product
9 comprises computer-executable instructions which, when executed by a processor,
10 implements the following:

11 attempting a connection for each connection request received by the server
12 computer system from said one or more client computer systems;

13 for each connection request that the server computer system cannot
14 currently handle, placing the connection request in a backlog queue;

15 monitoring the backlog queue;

16 determining that the backlog queue is being used;

17 resetting one or more connection sockets upon notification that the backlog
18 queue is being used.

19
20
21
22
23
24

20 18. The computer program product in accordance with Claim 17, further
21 comprising computer-executable instructions for mapping each connection request to a
22 corresponding listen socket, wherein each listen socket has a corresponding backlog queue.

1 19. The computer program product in accordance with Claim 17, wherein the
2 computer-executable instructions for placing the connection request in a backlog queue
3 comprise computer-executable instructions for placing the request in the backlog queue
4 corresponding to the listen socket that the connection request mapped to.

5

6 20. The computer program product in accordance with Claim 17, wherein the
7 computer-executable instructions for attempting a connection for each connection request
8 received by the server computer system from said one or more client computer systems
9 comprises at least portions of a Winsock module.

10

11 21. The computer program product in accordance with Claim 17, wherein the
12 computer-executable instructions for resetting one or more connection sockets upon
13 notification that the backlog queue is being used comprise computer-executable
14 instructions for performing the following:

15 identifying any connection sockets that have connections but no received
16 request data;
17 disconnecting the identified connection sockets.

18

19 22. The computer program product in accordance with Claim 17, further
20 comprising computer-executable instructions for performing the following:

21 specifying a grace period between the time the backlog queue is determined
22 to be used and the time one or more connection sockets are reset to allow the server
23 computer system to empty the backlog queue, wherein the resetting of the one or

1 more connection sockets is performed only if the backlog queue still has entries
2 after the grace period.

3
4 23. The computer program product in accordance with Claim 17, wherein the
5 computer-executable instructions for attempting a connection for each connection request
6 received by the server from said one or more clients comprise computer-executable
7 instructions for establishing a connection.

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

1 24. In a networked environment, wherein one or more client computer systems
2 make requests for information from a server computer system, the server computer system
3 providing information in response to the requests from the one or more client computer
4 systems, the server computer system having one or more listen sockets, each listen socket
5 having a backlog queue for queuing connection requests that the server computer system
6 cannot currently handle, a method of reducing denials of service even though the server
7 computer system is experiencing a denial of service attack, the method comprising:

8 attempting a connection for each connection request received by the server
9 computer system from said one or more client computer systems using a
10 Winsock()AcceptEx() module;

11 mapping each connection request to a corresponding listen socket;

12 for each connection request that the server computer system cannot
13 currently handle, placing the connection request in the backlog queue
14 corresponding to the listen socket that the connection request mapped to;

15 monitoring the backlog queue using a Winsock()select() module;

16 determining that the backlog queue is being used by detecting that the
17 Winsock()select() module has returned;

18 identifying any connection sockets that have connections but no received
19 request data using a Winsock()getsockopt() module; and

20 disconnecting the identified connection sockets.

21
22 25. The method in accordance with Claim 24, further comprising:
23 specifying a grace period between the time the backlog queue is determined
24 to be used and the time the identified connection sockets are disconnected, wherein

1
the disconnection is performed only if the backlog queue still has entries after the
2 grace period.
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24